多基站下基于 DRL 的 RAN 切片资源分配 *

马英洪，江凌云†

(南京邮电大学 通信与信息工程学院，南京 210003)

摘　要：在第五代移动通信中，网络切片被用来为各种业务提供一个最佳的网络。针对多基站下的 RAN 切片场景，以往的资源分配方法在切片的数量发生变化时无法满足切片的需求而且只适用于特定的场景，针对这个问题，提出了一种实现最佳资源分配且与切片数无关的方法。该方法先利用 Ape-X 方法(一种 DRL 方法)将资源分配给切片，再经过切片到基站的资源映射和用户资源分配来满足用户的需求。仿真结果表明，所提出的方法能够根据切片的状态和需求分配资源，分配了必要数量的 RB 以满足切片的需求而且不受切片数量变化的影响。同时，该方法也具有很高的通用性能和扩展性。

关键词：多基站；网络切片；深度强化学习；无线接入网；资源分配

中图分类号：TN929.5　　doi: 10.19734/j.issn.1001-3695.2021.12.0706

RAN slice resource allocation in multi base stations based on DRL

Ma Yinghong, Jiang Lingyun†

(*College of Telecommunications & Information Engineering*, *Nanjing University of Posts & Telecommunication*, *Nanjing* 210003, *China*)

Abstract: In the fifth generation mobile communication, network slicing is used to provide an optimal network for various services. For the RAN slice scenario under multi base stations, the previous resource allocation methods can not meet the demand of slices when the number of slices changes, and are only suitable for specific scenarios. To solve this problem, this paper proposes a method to achieve the best resource allocation independent of the number of slices. This method first uses Ape-X method (a DRL method) allocate resources to slices, and then meet the needs of users through the resource mapping from slices to base stations and user resource allocation. The simulation results show that the proposed method can allocate resources according to the state and demand of slices, allocate the necessary number of RBs to meet the demand of slices, and is not affected by the change of the number of slices. At the same time, this method also has high general performance and scalability.

Key words: multi base station; network slice; deep reinforcement learning; radio access network; resource allocation

0　引言

第五代移动通信(5G)网络作为解决日益增长的移动数据通信需求的解决方案引起了人们的关注。5G 改善了第四代移动通信(4G)网络中未得到适当解决的一些领域，例如更高的数据速率、更低的端到端(End to End, E2E)延迟、更高的可靠性和大量的设备连接[1]。此外，5G 中的服务类型也越来越多样化，例如需要高数据速率和低延迟的虚拟现实(Virtual Reality, VR)，以及需要大量设备连接和低延迟的工厂自动化[2]。传统的通信网络主要用来服务单一的移动宽带业务，无法适应未来 5G 多样化的业务场景。因此，为了在同一个物理网络基础设施上同时支持多种具有不同性能要求的业务场景，满足多样化的业务需求，网络切片技术应运而生。5G 使用网络切片(network slice)技术，以切片为单位提供适合各种服务的网络[3]。切片设置了吞吐量、延迟和可靠性等要求。为了满足这些需求，网络资源被分配到切片上。网络切片通常包括接入网切片(包括无线接入和固定接入)和核心网切片[4]。其中，无线接入网(Radio Access Network，RAN)必须要面对资源短缺的问题。而且，在实际分配无线资源的时候，切片的状态会不断变化，比如切片中的用户数量、业务的到达率以及用户的分布等。因此，需要一种能够根据切片状态有效地分配无线资源同时能够满足切片要求的方法[5,6]。

文献[7~12]是几种在单基站环境下的无线资源分配方法。文献[7]中的方法通过从没有需求的切片中分配资源来满足切片的需求。但是，没有评估资源分配的利用率。存在向切片分配过多资源的可能性。文献[8]提出了一种考虑切片需求和资源利用率的方法。但是无法对每个切片实现资源隔离，会受到其他切片的负载的影响。在文献[9]中采用了一种最早截止时间优先(Earliest Deadline First，EDF)调度策略用于无线资源分配，可以使得在高负载下满足切片对于时延的要求，但是会严重影响对于吞吐量有需求的切片的性能。文献[10]提出了一种在 C-RAN 场景下基于在线学习的网络切片虚拟资源分配算法，以最大化平均网络切片和速率为目标，同时考虑到了平均网络切片约束以及网络平均回传链路带宽消耗约束。但是没有考虑到切片的资源利用率。在文献[11]中的方法，利用了深度强化学习，同时考虑到了切片的满意度和资源利用率，而且实现了切片间的资源隔离。但是，仅评估了特定的场景。5G 中假设了各种服务场景，文献[11]中的方法可能并不适用。文献[12]提出了一种基于在线双向拍卖的网络切片资源调度机制。该机制能够在为高优先级切片服务的同时可以保证低优先级切片用户的 QoS 需求。但是没有考虑到切片间的资源隔离问题。

收稿日期：2021-12-31；**修回日期：**2022-03-10　**基金项目：**江苏省重点研发计划“智能网络服务平台及智慧城市应用关键技术研发”资助项目(BE2020084-4)；国家电网有限公司-科研资助项目(DSY2021-005)

作者简介：马英洪(1997-)，男，吉林松原人，硕士研究生，主要研究方向为 5G 网络和切片资源分配；江凌云(1971-)，女(通信作者)，安徽安庆人，副教授，硕导，硕士，主要研究方向为下一代网络(jiangly@njupt.edu.cn).

以上的文献研究的都是在单基站下的无线资源分配方法，在多基站环境下的分配情况会不同。文献[13~15]是几种在多基站环境下的资源分配方法。在文献[13]中，研究了 RAN 中混合业务的动态网络切片策略，考虑到了用户对时延和速率的 QoS 需求。但是对切片数量的设置不够灵活。文献[14]中的算法考虑到了基站的回传容量，对不同切片的用户的时延和数据速率的需求也满足的很好。但是在低负载的情况下资源的利用率不高。文献[15]设计并实现了一个两层网络虚拟化基板(Network Virtualization Substrate，NVS)算法，基于切片的优先级和可实现速率实现切片的资源调度。在这种方法中切片在各个基站上均匀分配资源，可能会导致在有的基站上分配资源过多，在有的基站上分配过少。从而导致用户的满意度和资源利用率不够好。

针对上述文献中的问题，本文提出了一种在多基站环境下使用 DRL 分配满足切片要求的无线资源的方法。而且，即使切片的数量发生变化，所提出的方法也能为各个切片分配资源以满足用户的 QoS 需求。

1　系统模型

1.1　网络模型

本文考虑 5G 基站 k，$k \in K$，K 是一个基站集。基站为切片 s 提供服务，$s \in S$，S 是切片集。为了满足用户的服务质量(Quality of Service，QoS)需求，必须在切片之间分配无线资源。在 5G 系统中，无线资源用资源块(Resource Block，RB)来表示。RB 是基于时域和频域划分的资源单位。一个 RB 由 12 个子载波组成，每个子载波的间隔为 15kHz。时域以传输时间间隔(Transmission Time Interval, TTI)为单位，TTI 的长度为 1ms。TTI 是调度 RB 的最小单位，在每个 TTI 将 RB 分配给用户。

基站 k 将 RB 分配给用户以满足用户的需求，用户 u 在一个 RB 上的可实现速率计算为：$r_{k,u} = B\log_2(1+\frac{p_k g_{k,u}}{N_0 B})$。其中，B 表示一个 RB 的带宽，即 15kHz×12=180kHz。p_k 表示基站 k 的发射功率，$g_{k,u}$ 表示用户 u 在基站 k 一个 RB 上的信道增益，N_0 表示噪声功率谱密度。用户 u 在不同基站上的 RB 的可实现速率不同，因此，用户按照不同基站上 RB 的可实现速率由大到小设立等级，用户 u 对基站 k 的等级记为 $\delta_{u,k}$，可表示基站 k 对用户 u 的重要性。

不同切片的用户的 QoS 需求不同，在本文中，将需求分为两种：吞吐量需求和时延需求。对于需要高吞吐量的用户在每个 TTI 为其分配 RB；对于有时延需求的用户在数据包到达时分配 RB。假设用户 u 对吞吐量的需求为 R_u，也就是对数据速率的需求。用户 u 对时延的需求为 T_u，用户 u 的数据包大小为 p_u。因此，用户 u 对数据速率的需求 $R_u = {}^{p_u}\!/_{T_u}$。由于 TTI 是最小的时间调度单位，因此，用户 u 的数据速率需求 R_u 需要转换成 1ms 上的速率需求。此外，本文将切片的需求设为用户的 QoS 需求，同一切片内的用户的 QoS 需求相同。

本文定义了两个指标：网络切片需求满意度(NSDS, network slice demand satisfaction)和 RB 利用率(RBUR, RB usage ratio)。第一个指标 NSDS 用来衡量网络是否满足了该业务的需求。NSDS 表示为：$NSDS = \frac{\sum_{k=1}^{|k_s|} u_k}{|k_s|}$。$k_s$ 表示切片 s 中的用户数量。u_k 表示用户是否满足了切片需求。

$$u_k = \begin{cases} 1，满足切片需求 \\ 0，未满足切片需求 \end{cases} \tag{1}$$

NSDS 越接近 1，就说明切片中用户的 QoS 需求满足的越好，就可以为服务提供更合适的切片。RBUR 是一个用来衡量资源利用率的指标，RBUR 表示为 $RBUR = {}^{URB}\!/_{ARB}$。URB 表示消耗的 RB 数量，ARB 表示分配给切片的 RB 数量。RBUR 越接近 1，RB 的利用率就越高，资源利用率就越高。

当 NSDS 很低而 RBUR 很高时，分配给切片的 RB 都被消耗掉，切片中有的用户没有满足切片的需求。因此，给切片分配更多的 RB 可以改善 NSDS。通过最大化 NSDS 和 RBUR，可以实现以最少的 RB 数量满足切片的需求。

1.2　问题模型

RB 分配问题可以分成两步：切片间资源分配和切片内资源分配。切片间资源分配问题包括分配 RB 给切片和切片到基站的资源映射。切片间资源分配问题建模如下：

$$\begin{aligned} &\max \sum_{i=1}^{S} NSDS_i \times RBUR_i \\ &s.t.: \sum_{i=1}^{S} ARB_i \le AllRB \\ &\quad\ \sum_{i=1}^{S} R_{s,k} \le L_k, \forall k \in K \end{aligned} \tag{2}$$

目标是最大化切片的资源利用率 RBUR 和切片满意度 NSDS 的乘积，以最小的 RB 分配满足切片需求。式(2)中第一个约束表示所有切片所分配的 RB 数量不能超过基站所拥有的 RB 总数。第二个约束表示在每个基站下，所有切片获得的 RB 的总数量不能超过该基站所拥有的 RB 数量。

在确定了切片在每个基站上的资源分布后，网络切片控制器执行切片内资源分配。切片中的用户优先接入 RB 等级最高的基站，这样可以最大化满足用户的 QoS 需求。在基站 k 下，设定分配给切片 s 的 RB 集为 $M_{s,k}$，切片内资源分配问题建模如下：

$$\begin{aligned} &\max_{A} \{U_{s,k}\}, \forall s \in S \\ &s.t.: \sum_{n \in M_{s,k}} \sum_{u \in U_{s,k}} a(u,n) \le R_{s,k} \\ &\sum_{u \in U_{s,k}} a(u,n) \le 1, \forall n = 1,...R_{s,k} \end{aligned} \tag{3}$$

在式(3)中，A 是分配矩阵，如果将第 n 个 RB 分配给第 u 个用户，则元素 $a(u,n)$为 1，否则为 0。$U_{s,k}$ 表示第 s 个切片在第 k 个基站上的效用，本文考虑 $\alpha-utility$ [16]，如式(4)所示。

$$U_{s,k} = \sum_{u \in U_{s,k}} \begin{cases} \dfrac{(\sum_{n \in M_{s,k}} a(u,n) r_{u,n})^{1-\alpha}}{1-\alpha}, \alpha \ne 1 \\ \log(\sum_{n \in M_{s,k}} a(u,n) r_{u,n}), \alpha = 1 \end{cases} \tag{4}$$

本文选择 α=1，即比例公平方式。在式(3)中，第一个约束表示在基站 k 下，每个切片内的用户分配的 RB 的总数不能超过该切片可用的资源。第三个约束确保一个 RB 只能分配给一个用户。

2　算法设计

2.1　方法概述

在 RAN 切片中，需要一种以最小 RB 分配满足切片需求的方法，该方法与切片数量无关。因此，本文提出了一种利用 Ape-X[17]的灵活 RB 分配方法。由于使用了 Ape-X，learner 学习的模型包括基于每个 actor 收集的各种经验。因此，当切片数量变化时无须重新训练模型即可进行 RB 分配。在现有方法中，agent 控制的切片数量是固定的，因此如果训练和评估之间的切片数量不同，则需要重新训练模型。在所提出的方法中，一个 agent 将 RB 分配给一个切片，当有多个切片时，agent 会被多次调用。 这种设计实现了与切片数量无关的 RB 分配。在每次 RB 分配后，网络切片控制器会更新切片在每个基站上的资源配置，以便 BS 可以在每个调度时期适应系统状态。

此外，代理学习以最少所需的 RB 数量分配来满足切片

的需求，从而最大化满足需求的切片数量，同时提高 RB 的利用效率。在本文中，为每种服务类型定义一个切片。当切片中的用户数变为 1 或更多时生成切片，在用户数变为 0 时终止切片。所提出的算法的流程图如图 1 所示。

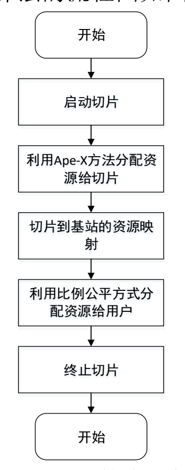


图 1　所提出的算法的流程图

Fig. 1　Flow chart of the proposed method

2.2　利用 Ape-X 的 RB 分配

所提出的方法采用 Ape-X 方法，将分布式学习应用于 DRL。其中，一个 actor 控制一个切片，当切片的数量发生变化时，actor 的数量也随之变化，切片的数量没有限制，分配 RB 不依赖于切片的数量。所提出的方法可以灵活设置切片数量和分配 RB。learner 学习一种策略，该策略以最少的 RB 数量满足切片的需求。分配方法的架构如图 2 所示。由于切片数量没有限制，所以有 1 到 N 个切片，每个切片包含多个用户。除了管理切片之外，网络切片控制器还桥接了基站和 actor。actor 是 Ape-X 代理，和切片的数量相同。由于 actor 使用了 learner 训练的策略，所有 actor 都具有相同的控制策略。在每个资源调度时期，执行 RB 的分配操作。基站收集每个切片的状态信息，包括切片中的用户是否满足了 QoS 需求以及切片的资源利用率等信息。随后，基站将切片状态通知给网络切片控制器。网络切片控制器根据切片状态生成状态和奖励，并将其传递给切片对应的 actor。actor 根据策略生成动作，把动作输出到网络切片控制器。当网络切片控制器接收到每个切片的动作时，它会计算分配给它们的 RB 数量。在学习过程中，奖励、状态和动作作为经验传递给回放记忆 (replay memory)。在切片级资源更新后，资源需要映射到所有基站上。2.3 节介绍了基站资源更新的算法流程。在进行基站级资源更新后，网络切片控制器通知各基站每个切片在其上的资源分布。各个基站为每个切片中的用户分配 RB，以满足用户的吞吐量和时延需求。

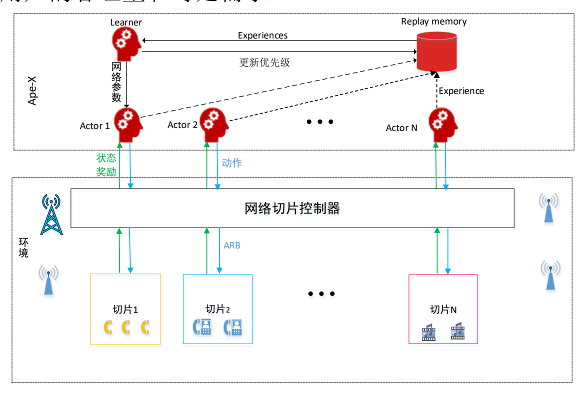


图 2　RB 分配方法的架构

Fig. 2　Architecture of RB allocation method

本文利用深度强化学习解决切片间资源分配问题，作出最优决策。在下面详细介绍强化学习架构中各个要素的内容。

状态是代理确定动作的重要因素。如果状态被设计为尽可能地消除不确定元素，则学习结果会更好。基于这一点，本文将学习 RB 分配时的状态分为表 1 中给出的三种类型。这三种类型分别是 NSDS 相关、RBUR 相关和切片状态。首先，NSDS 相关，对于代理识别切片的需求很重要。第二个，RBUR 相关，帮助代理识别切片的 RB 分配情况。第三个是为了解决状态的模糊性。

表 1　RB 分配的状态设计

Tab. 1　State design of RB allocation

状态类型	值
NSDS 相关	NSDS
	吞吐量需求
	时延需求
RBUR 相关	RBUR
	分配 RB 的数量
切片状态	到达数据包的数量
	传输数据包的数量
	缓冲区中数据包的数量

动作是代理对环境执行的控制。所提出的方法为每个切片分配 RB。由 actor 输出的动作记为 a，a 可以为负，0 或正。a 为负表示分配给切片的 RB 数量减少，a 为 0 表示分配给切片的 RB 数量不变，a 为正表示分配给切片的 RB 数量增加。动作 a 的取值范围为[-0.8,-0.6,-0.4,-0.2,0,0.2,0.4,0.6,0.8]，动作 a 为九个维度。在 t+1 时刻，分配给切片的 RB 数量可以计算为 $ARB_{t+1}=ARB_t\times(1+a)$。计算出的 ARB 不考虑分配给其他切片的 RB。然而，由于基站具有有限数量的 RB，分配给切片的 RB 总数可能会超过基站的所有 RB。因此，网络切片控制器根据算法 1 调整分配给每个切片的 RB 数量。对每个切片计算时间 t 处分配的 RB 数和 NSDS 的乘积，按照从小到大的顺序进行排列。按照这个次序为切片分配资源。这使得需要较少 RB 的切片或具有较小 NSDS 的切片在时间 t+1 时优先分配。这样可以避免需要更多 RB 的切片占用多余的资源，提高了资源利用率。另一方面，可以改善切片的 NSDS。

奖励向代理表明一个动作对于一个状态是好是坏。在所提出的算法中，目标是以最少的 RB 分配满足切片的需求，最大化 NSDS 和 RBUR 是学习的目标。奖励 r 被设计为

$$r=\begin{cases}1,(ARB=0,Buff=0)\\0,(ARB=0,Buff>0)\\0,(ARB>0,Buff=0)\\NSRS\times RBUR,(ARB>0,Buff>0)\end{cases}\tag{5}$$

ARB 是分配给切片的 RB 数量，Buff 是存储在缓冲区中的数据包的数量。当缓冲区为空时，不需要分配 RB，因此如果 ARB 为 0，则 r 为 1；ARB 不为 0，r 为 0。当缓冲区有数据包而 ARB 为 0，即切片有需求但没有被分配 RB，r 为 0。当数据包存储在缓冲区中并且 ARB 不为 0 时，NSDS 和 RBUR 会随着分配的 ARB 而变化。因此，r 是根据 NSDS 和 RBUR 计算得出的。

DQN 的框架如图 3 所示，其中包含两个神经网络，$Q(s,a;\theta)$ 代表预测 Q 网络，θ 表示预测神经网络的参数，用于评估当前状态-动作对的价值；$Q(s,a;\theta^*)$ 代表目标 Q 网络，用于计算目标值，θ*表示目标神经网络的参数。损失函数 $L(\theta)$ 计算为：$L(\theta)=E[(y_t-Q(s_t,a_t;\theta))^2]$，$y_t=r_{t+1}+\gamma\max_{a\in A}Q(s_{t+1},a;\theta^*)$。$y_t$ 是目标值，$y_t-Q(s,a;\theta)$ 即为 TD(Temporal Difference)误差 δ_t。利用损失函数计算梯度。计算表达式为

$$\frac{\partial L(\theta)}{\partial\theta}=\frac{\partial E[(y_t-Q(s_t,a_t;\theta))^2]}{\partial\theta}\tag{6}$$

本文使用 RMSProp 优化算法更新网络参数，每经过 M 步迭代，复制预测网络的参数 θ 给目标网络参数 θ*。

Ape-X 使用了 DDQN[18]，利用预测网络选择动作，目标网络计算 Q 值，降低了高估。此外，Ape-X 还使用了多步引导目标(multistep bootstrap target)[19]，即考虑多步的奖励，使学习的效果更好。因此，目标值 y_t 变为

$$y_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n Q(s_{t+n}, \arg\max_{a\in A} Q(s_{t+n}, a; \theta); \theta^*) \quad (7)$$

由于采用了决斗网络[20]，$Q(s_t,a_t;\theta)$ 被划分为状态价值函数 $V(s_t;\theta)$ 和优势函数 $Adv(s_t,a_t;\theta)$。状态价值函数仅和状态 s_t 有关，与动作 a_t 无关。优势函数同时和状态 s_t 和动作 a_t 都有关。Q 值表示如下：

$$Q(s_t,a_t;\theta) = V(s_t;\theta) + (Adv(s_t,a_t;\theta) - \frac{1}{|A|}\sum_{a\in A} Adv(s_t,a;\theta)) \quad (8)$$

其中，A 表示动作集，|A|是动作集的维度。actor 与环境交互产生经验 (s,a,r,s')。本文采用ε-greedy 策略，actor 以ε的概率选择动作值最高的动作，以概率 1-ε选择随机动作。actor 根据其网络参数计算经验的 TD 误差，根据 TD 误差设置经验的优先级 p_k，$p_k = |\delta_k|$，k 是经验的编号。经验 (s,a,r,s',p_k) 被存入经验回放中，预测网络和目标网络使用优先级采样代替随机采样，从经验回放中提取经验。经验的采样概率为 $p_k / \sum_k p_k$，每次抽取 batch 个样本。

在更新网络参数后，计算 batch 个样本的 TD 误差和优先级，然后更新经验回放中这些样本的优先级。随着学习的进行，动作值被更新，旧经验的准确性降低，存储在经验回放中的旧经验的优先级被更新为低。当经验回放被装满时，删除旧经验。

每隔 N 步，actor 从预测网络处复制经过训练的参数，以将其参数更新为最新版本。这样，通过优先学习具有较大 TD 误差的经验，同时执行分布式学习来加速学习，从而提高了学习效率。

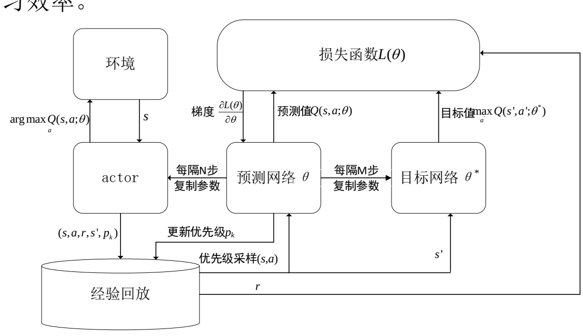


图 3　DQN 框架

Fig. 3　DQN framework

图 4 显示了所提出方法的神经网络结构。该网络结构有一个输入层、几个隐藏层和一个输出层。因为状态维度被设计为输入维度，所以输入层是八维的。输出层的维度为 9 维，和动作维度相同。隐藏层有 128 个神经元。第 4 个隐藏层分支到状态值和优势函数。使用这些规格进行了 2×10⁶ 步训练。训练大约需要 2 天。

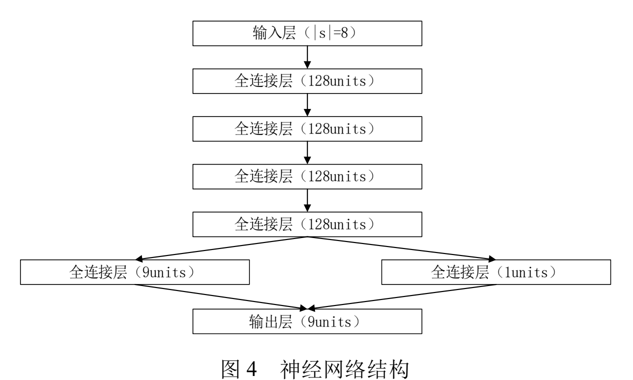


图 4　神经网络结构

Fig. 4　Neural network structure

算法 1　切片的 RB 分配

输入：CRB_{t+1}^s 是在 t+1 时刻计算出的分配给切片 s 的 RB 数，ARB_t^s 是在 t 时刻分配给切片 s 的 RB 数，$NSRS^s$ 是切片 s 的网络切片需求满意度，S 是切片集，$AllRB$ 是所有基站所拥有的全部资源。

输出：ARB_{t+1}^s 是在 t+1 时刻分配给切片 s 的 RB 数。

1)开始：
2)for　$s \in S$
3)　W[s] = $ARB_t^s \times NSRS^s$
4)end for
5)remainRB=$AllRB$
6)for $s \in S$ #按照 W 的升序提取 s
7)　if　CRB_{t+1}^s <remainRB
8)　　$ARB_{t+1}^s = CRB_{t+1}^s$;
9)　　remainRB = remainRB - ARB_{t+1}^s
10)　else
11)　　ARB_{t+1}^s = remainRB
12)　　remainRB = 0
13)　end if
14)end for
15)结束

2.3　切片到基站的资源映射

2.3.1 基站资源更新

在切片资源更新后，资源必须映射到所有基站上。算法 2 描述了基站资源更新过程。基站资源更新依赖于基站对特定切片的权重。切片中的用户在不同基站上的 RB 等级 $\delta_{u,k}$ 不同，可以得到切片 s 在基站 k 上的等级：$\bar{r}_{s,k} = \sum_{u=1}^{|U_s|} \delta_{u,k}, \forall u \in U_s$。因此，切片 s 在基站 k 上的权重可以表示为：$r_rat_{s,k} = \frac{\bar{r}_{s,k}}{\sum_{k=1}^{K} \bar{r}_{s,k}}$，权重 $r_rat_{s,k}$ 表示了基站 k 对切片 s 的重要性。在每个调度时间，这些权重都会更新，以更新基站资源。切片 s 在基站 k 上占有的资源计算为：$R_{s,k} = ARB_s \times r_rat_{s,k}$。计算出的 $R_{s,k}$ 没有考虑到每个基站的容量限制，所有切片在基站 k 上分配的 RB 总数可能会超出基站 k 的容量。因此，需要调整切片在每个基站上所分配的资源。遍历所有基站，将分配资源量超出基站容量的基站添加进待重新分配队列 Q_BS。

对于属于 Q_BS 队列的基站 k，先确定属于该基站 k 的用户。根据用户的 RB 等级得到切片 s 在基站 k 上的用户集合 $U_{s,k}$，$k = \max\{\delta_{u,k}(k)\}$，用户 u 优先选择 RB 等级最高的基站接入。切片 s 在基站 k 上的速率需求为：$W_{s,k} = \sum_{u=1}^{|U_{s,k}|} R_u$，$R_u$ 表示用户 u 的需求速率。因此，切片 s 在基站 k 上的权重可以计算为：$\omega_{s,k} = \frac{\bar{r}_{s,k} \times W_{s,k}}{\sum_{s=1}^{|S|} \bar{r}_{s,k} \times W_{s,k}}$，由此得到，在基站 k 上分配给切片 s 的 RB 数为：$R_{s,k} = L_k \times \omega_{s,k}$，$L_k$ 为基站 k 所拥有的 RB 总数。

在对基站的资源进行重新分配后，有的切片分配的资源总数可能会低于 ARB，需要将基站剩余的资源分配给切片。首先检查每个基站的剩余资源量以及分配资源不够的切片。还有剩余资源的基站队列设为 Q_reBS，分配 RB 数不够的切片队列设为 Q_slice。遍历基站队列 Q_reBS，比较切片队列 Q_slice 中的所有切片在该基站上的权重 $\omega_{s,k}$，按照权重由大到小的次序，将基站的剩余 RB 资源分配给切片直到满足切片的 RB 需求或基站的资源全部被分配。

算法 2　基站资源更新

输入：分配给切片 s 的 RB 数 ARB^s，切片 s 在基站 k 上的等级 $\bar{r}_{s,k}$，用户 u 的请求速率 R_u，切片集 S，基站集 K，切片缺少的 RB 数 $reRB$，基站 k 的 RB 容量 L_k，基站 k 剩余的 RB 资源 Lk_remain，待重新分配切片队列 Q_slice，有剩余资源的基站队列 Q_reBS，在 t+1 时刻分配给切片 s 的 RB 数 ARB_{t+1}^s。

输出：$R_{s,k}$ 是基站 k 分配给切片 s 的 RB 数。

开始：初始化 Q_slice，Q_reBS 队列

Step1：初步资源映射

for $s \in S$

　for $k \in K$

　　$R_{s,k} = ARB^s \times \frac{\bar{r}_{s,k}}{\sum_{k=1}^{K} \bar{r}_{s,k}}$；

　end

end

for $k \in K$

　$BS_exceed^k = \sum_{s=1}^{|S|} R_{s,k}$；

　if $BS_exceed^k > L_k$

　　for $s \in S$

　　　获取一个用户集 $U_{s,k}$，$k = \max\{\delta_{u,k}(k)\}$

　　　$W_{s,k} = \sum_{u=1}^{|U_{s,k}|} R_u$；

　　　$\omega_{s,k} = \frac{\bar{r}_{s,k} \times W_{s,k}}{\sum_{s=1}^{|S|} \bar{r}_{s,k} \times W_{s,k}}$；

　　　$R_{s,k} = L_k \times \omega_{s,k}$；

　　end

　end

　if $BS_exceed^k < L_k$

　　$L_{k_remain} = L_k - BS_exceed^k$；

　　Q_reBS.add(k);

　end

end

Step2：基站剩余资源分配

for $s \in S$

　$sumRB^s = \sum_{k=1}^{|K|} R_{s,k}$；

　if $sumRB^s < ARB^s$

　　$reRB^s = ARB^s - sumRB^s$；

　　Q_slice.add(s);

　end

end

for $k \in Q_reBS$

　按照切片 s，在基站 k 上的权重 $\omega_{s,k}$ 由大到小进行排序，得到 $Rank$

　if len(Q_slice)==0

　　break;

　end

　for $s \in Rank$

　　if $L_{k_ramain} \geq reRB^s$

　　　$reRB^s$=0;

　　　Q_slice.delete(s);

　　　$L_{k_remain} = L_{k_remain} - reRB^s$；

　　else

　　　$reRB^s = reRB^s - L_{k_remain}$；

　　end

　　if $L_{k_remain} == 0$

　　　Q_reBS.delete(k);

　　　break;

　　end

　end

end

2.3.2 用户资源分配和连接控制

为了解决切片内资源分配的问题，本文提出了一个迭代的解决方案。分配矩阵 A 最开始为空，然后通过迭代将 RB 分配给用户。在这里定义一个增益因子：$g_u^{(i+1)} = \log(\hat{R}_u^{(i)} + r_{u,\hat{n}}) - \log(\hat{R}_u^{(i)})$。$\hat{R}_u^{(i)}$ 表示用户 u 已分配的速率，$r_{u,\hat{n}}$ 表示用户 u 在 $\hat{n}$ 上的可实现速率，$\hat{n}$ 是还未分配的 RB。如果用户 u 具有最大的增益因子而且该用户的数据速率需求还未得到满足，则将 $\hat{n}$ 分配给该用户。然后进入下一次迭代，当所有的 RB 被分配完成或者所有用户的需求都被满足时，迭代停止。

当所有基站下的所有切片的迭代分配过程完成后，检查是否所有用户都满足了需求以及切片的资源是否都被利用了。因为可能接入一个基站的用户过多而分配的资源不够，接入其他基站的用户少而分配的资源很多。因此，为了充分利用资源以及满足用户需求，将切片在其他基站上的资源分配给未满足需求的用户。对于未满足需求的用户，按照 RB 等级次序，检查其他基站是否有剩余的 RB 资源，如果有，就将用户连接到下一个基站，根据比例公平方式迭代将该基站剩余的 RB 资源分配给重新接入该基站的用户们，直到切片中的用户都满足了需求或分配给切片的资源全部被利用就停止迭代。

3　仿真结果

3.1　仿真概述

所提出的方法是根据它是否达到满足切片需求的最小 RB 分配来评估的，而且不受切片数量变化的影响。在所提出的方法中，必须首先训练 RB 分配的模型。然后使用经过训练的模型评估所提出的方法。评估分为三种。第一种表明所提出的方法在所创建的特定场景中适当地实现了 RB 分配。第二种展示了基于随机生成的多个场景评估模型的通用性能。在第三种类型中，本文评估切片数量与性能之间的关系，并描述了所提出方法的可扩展性。

3.2　训练

所提出方法中的模型从切片状态中学习最大化 NSDS 和 RBUR 的 RB 分配。在 5G 中，假设了各种类型的服务。因此，仿真场景是随机生成的，模型使用各种类型的服务进行训练。表 2 给出了用于训练的场景。表 3 给出了用于训练和评估的共同参数。每次仿真后都会生成一个新场景。切片的数量固定为三个切片，但在仿真过程中，切片的数量从 0 到 3 不等，因为切片的开始和结束时间不同。此外，每个切片的用户数量、数据包生成间隔和数据包大小都不同。切片的需求为吞吐量需求、时延需求或两者都有。根据长期演进 (Long Term Evolution, LTE) 规范，子载波间隔和 TTI 分别设置为 15 kHz 和 1 ms。系统带宽设置为 20 MHz，每个 TTI 的 RB 总数为 100。由于 RB 是根据规范分组的，因此 每个基站控制的 RB 数为 25。RB 分配的控制间隔为 1ms，和 TTI 一样。

表 4 给出了 Ape-X 的参数值，和[17]中的相同。actor 的数量是根据计算机的性能设置的参数。训练电脑的配置为 AMD 3700x CPU，32GB 内存，rtx 2070super 显卡。进行仿真的电脑为 3700x CPU，16GB 内存。本文使用一台计算机进行学习，六台计算机进行仿真。在一台计算机上运行 5 个仿真，而一个仿真有 4 个 actor，所以总共有 120 个 actor(6 计算机×5 个仿真×4 个 actor)。折扣因子决定了考虑未来奖励的折扣因子。在 RB 分配中，切片状态也会快速变化，因此根据状态变化快速分配 RB 是很重要的。因此，本文将折扣因子设置为 0.5 以最大化短期奖励。出于同样的原因，将 n 设为 1。

3.3　评估结果

3.3.1 RB 分配评估

在本节中，本文评估所提出的方法是否能在切片数量不同的场景中将 RB 分配给切片。所考虑的移动网络场景基于 5G 网络标准，所使用的参数汇总在表 3 和 5。在给定的 500m×500m 区域内，四个基站均匀分布。每两个相邻的 BS

保持 120m 的固定距离。路径损耗(Path Loss, PL)模型定义如下：PL(dB)=20log10(d)+20log10(f)-27.55。其中，d(以米为单位)和 f(以 MHz 为单位)分别表示用户-基站的距离和信道频率。基于 5G 切片类别，本文定义了 4 个切片：消息业务、应用程序、音频、视频，每个切片匹配一种业务。每个切片具有不同的用户数量、数据包长度和切片需求。通过设置每个切片的切片开始和结束时间来模拟切片数量的变化。在仿真中，切片数量从最少两个切片到最多四个切片不等。具体的参数如表 5 所示。切片中的用户数量随时间而变，如图 5 所示。

表 2　训练场景-Uniform[min, max]表示从 min 到 max 的均匀分布

Tab. 2　Training scenario-Uniform[min,max], represents a uniform distribution from min to max

参数	值
仿真时间	30s
基站数量	4
基站间距离	120m
最大用户数 mu	400
切片数量	4
切片的开始时间 st	Uniform[1,10][s]
切片的结束时间	st+Uniform[1,20][s]
切片中的用户数量	Uniform[1,200]，所有切片的用户总数不超过 mu
数据包生成间隔 pi	Uniform[1,1000][ms]
数据包大小 ps	Uniform[1,65535][Bytes]
吞吐量需求	tc×ps×1000/pi[Bps],tc=Uniform[0.8,1],当tc=0.8 时，对吞吐量没有需求
时延需求	dl=Uniform[3,1000][ms],当 dl=1000 时，对时延没有需求

表 3　训练和评估的共同参数

Tab. 3　Common parameters for training and evaluation

参数	值
区域大小	500×500[m²]
用户移动性	随机路点模型
基站发射功率	46dBm
子载波间隔	15kHz
TTI	1[ms]
系统带宽	20[MHz]
噪声功率谱密度	-174dBm/Hz
控制间隔	1[ms]

表 4　Ape-X 参数

Tab. 4　Ape-X parameters

参数	值
Actor 数量	120
目标网络更新间隔	2500[step]
网络参复制间隔	100[step]
训练 batch 大小	512
学习率 α	0.00025/4
折扣因子 γ	0.5
优化算法	RMSProp
多步 bootstrap 目标 n	1

本文中对比了两种方法。方法描述如下所示。

a)hard-slicing。hard 方法是一种将所有 RB 按切片数量划分的方法。切片 s 的 ARB 可以计算为：$ARB^s = AllRB / 切片数量$。

b)NVS(Network Virtualization Substrate)方法[18]。资源基于系统中切片的权重来分配，切片的权重计算如下：$\omega_s = \sum_{u=1}^{|U_s|} R_u, \forall u \in U_s$。定义为切片 s 中的所有用户的聚合数据速率请求。因此，分配给切片 s 的 RB 计算为：$ARB^s = AllRB \times \frac{\omega_s}{\sum_{s=1}^{|S|} \omega_s}$。

在 NVS 方法中，为每个切片提供的资源量在基站之间平均分配。本文所提出的方法以及 hard 方法采用 2.3 节所述的资源映射方式。

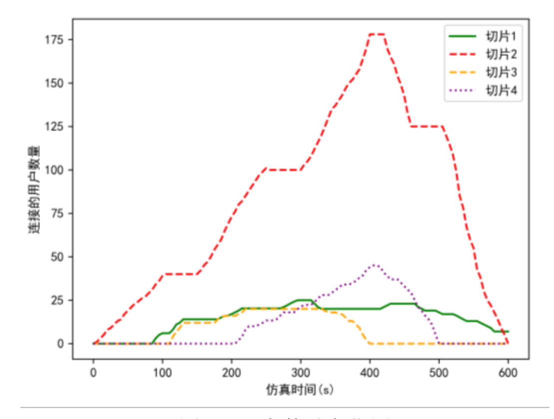


图 5　用户数量变化图

Fig. 5　Changes in the number of users

表 5　评估场景

Tab. 5　Evaluation scenario

参数	值			
仿真时间	600s			
基站数量	4			
基站间距离	120m			
用户的总数	290			
切片的数量	4			
切片编号	1	2	3	4
服务类型	消息业务	应用程序	音频	视频
用户数量	25	200	20	45
数据包到达率/包/s	Uniform[80,120]			
数据包大小/Byte	476	76	147	1055
吞吐量需求/Mbps	无	0.07	无	0.82
时延需求/ms	10	无	50	100
切片开始时间/s	0	0	100	200
切片结束时间/s	600	600	400	500

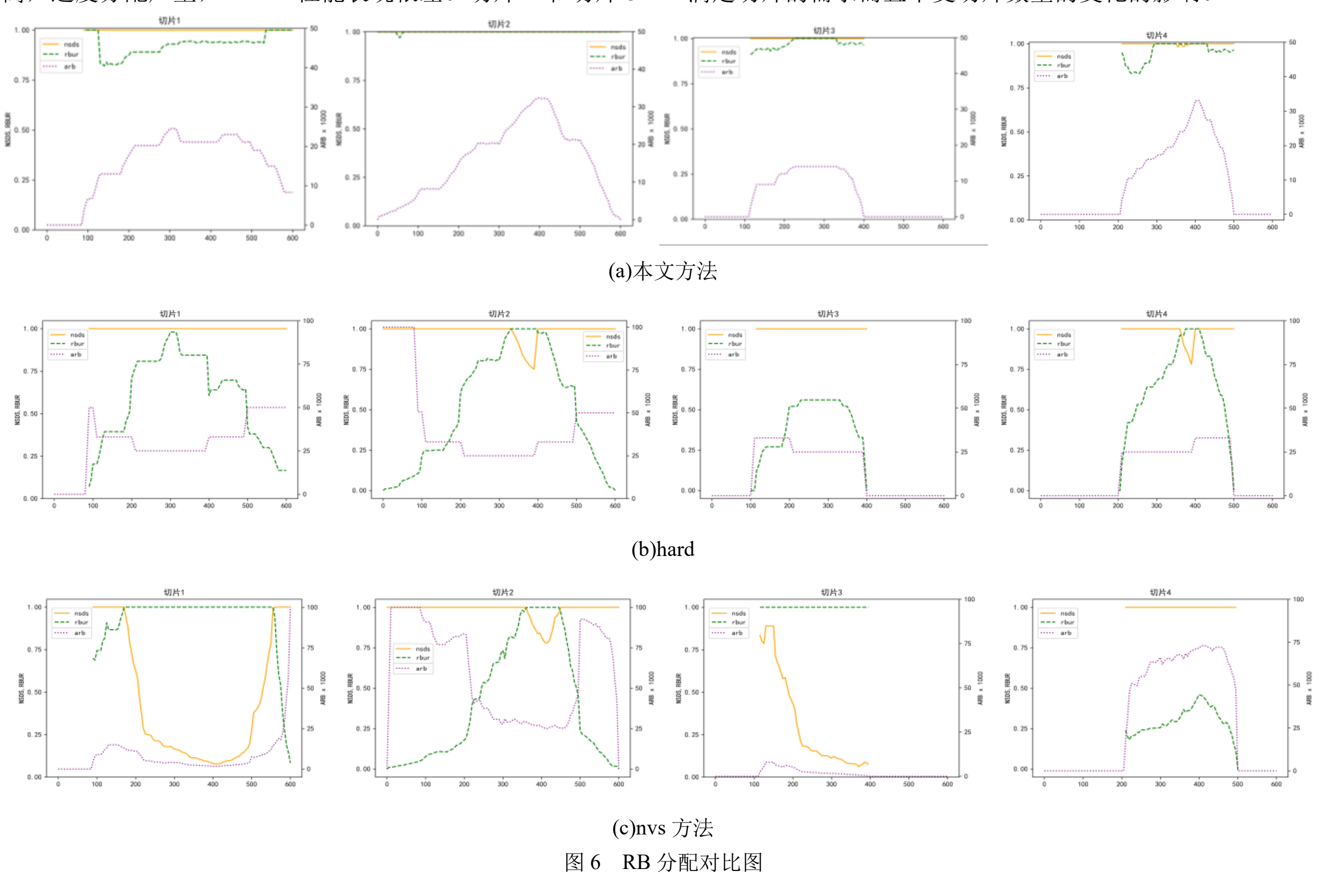
图 6(a)到 6(c)显示了 NSDS、RBUR、ARB 和仿真时间之间的关系。这里，RB 以一毫秒的间隔分配给切片，但 NSDS 和 RBUR 是以一毫秒为间隔测量的一秒的平均值，ARB 是 1s 的总 ARB(100×1000RB)。NSDS 的结果表明，所提出的方法几乎完全满足切片需求。在所提出的方法中，当切片的数量在 110s、205s、400s 发生变化时，NSDS 不会下降。基于这些结果，即使切片的数量发生变化，切片的性能需求也能得到满足。

在 hard 方法中，RB 分配和切片的数量有关，因此，对数据速率需求低的切片 1、3 总是能够满足切片需求。切片 2 中的用户数量多，在 320s 到 400s 之间时，切片 2 的用户数量不断增加，但是分配的 RB 数量没有增加，资源分配不足，导致 NSDS 降低。切片 4 需要高吞吐量，在 350s 到 400s 之间时，用户数量增加，对数据速率的需求也会增加，但是 ARB 不会变化，NSDS 下降。

在图 6(c)中，依据 nvs 方法，分配给切片的资源和切片的数据速率需求有关。数据速率需求越高，获得的资源就越多。因此，从图(c)中可以看出，在大约 90s 和 160s 之间，切片 1 的需求都能得到满足，在 160s 到 200s 之间，随着切片 2 的用户数量不断增加，切片 2 对于数据速率的需求也越来越高，切片 2 获得的资源也变多，导致切片 1 的 NSDS 下降。在 200s 和 500s 之间，切片 4 因为其高吞吐量需求，而占有了过多的资源。在 400s 左右时，切片 4 的用户数量达到最大，获得 RB 最多，其他切片的 NSDS 也达到最低。切片 3 由于其数据速率需求低而且用户数量少，因此获得的 RB 数量少，NSDS 很低，无法满足切片对于时延的需求。

根据 RBUR 结果，所提出方法的 RBUR 约为 0.75 或更高。这表明 RB 的过度分配低于 25%。在 hard 方法中，由于将 RB 均匀地分配给切片，而不管切片的状态和需求，因此 RBUR 的表现很差，在每个切片中都存在过度分配，如图(b)所示。对于 nvs 方法，切片 2 和切片 4 的数据速率需求高，过度分配严重，RBUR 性能表现很差。切片 1 和切片 3 对数据速率的需求低，导致分配的 RB 不足，所分配的 RB 都被利用了，因而资源利用率比较高。

综上所述，基于 NSDS 和 RBUR 的结果，所提出的方法对比其他方法在 NSDS 和 RBUR 两个指标上表现更好，能够根据切片的状态和需求分配资源，分配了必要数量的 RB 以满足切片的需求而且不受切片数量的变化的影响。



(a)本文方法

(b)hard

(c)nvs 方法

图 6　RB 分配对比图

Fig. 6　RB allocation comparison chart

3.3.2 通用性能评估

在本节中，本文使用模拟各种服务的场景来评估所提出方法的通用性能。作为包括 DRL 在内的通用机器学习，可以通过仅训练特定数据来估计目标数据的最优解决方案。但是不能对其他未经训练的数据进行正确的估计。这称为过度适应环境。当只针对特定的时区、地点或服务时，这种模型是有效的。但是，如果模型只用于特定情况，则使用网络切片的优势就丧失了。这是因为在 RAN 中，会有很多的服务类型，而网络切片是一种将网络适用于各种服务的技术。

使用第 3.2 节中的随机生成场景评估所提出方法的通用性能。由于该场景随机确定用户数量、数据包生成间隔和切片需求，因此可以模拟各种业务类型。需要注意的是，并非所有生成的场景都可以通过现有服务来解释。本文使用与训练中不同的种子值来评估未经训练的场景，以显示模型的通用性能。

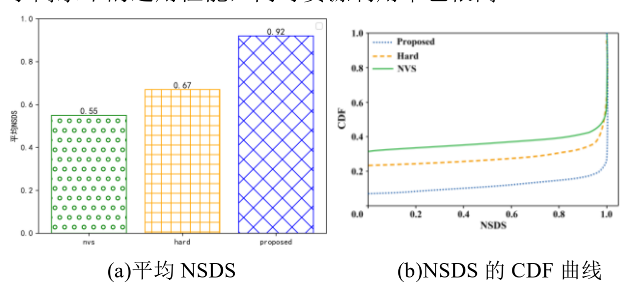
总共测试了 3000 个场景。评估指标为 NSDS 和 RBUR，是一秒内测量值的平均值。结果表示为累积分布函数 (CDF) 和测量数据的平均值。如果 NSDS 和 RBUR 较高，则表明它们在各种随机生成的场景中都较高。

图 7(a)和 7(b)显示了 NSDS 评估结果。所提出的方法的平均 NSDS 约为 0.92，几乎满足了切片需求。对比方法的平均 NSDS 都在 0.7 以下。由图 7(b)可以看出，hard 切片方法的 NSDS 在没有用户满足需求的 0.0 处以大约 28%分布，在所有用户都满足需求的 1.0 处以大约 50%分布。结果表明，切片中的一个用户有 28%的概率无法满足需求。对于 nvs 方法，切片中的一个用户有 31%的概率无法满足需求，切片中的用户的需求全部得到满足的概率是 50%。

所提出的方法的 NSDS 在 0.0 处以不到 10%分布，在 1.0 处大约 82%分布。结果表示切片中的一个用户无法满足需求的概率不到 10%，切片中的用户的需求全部被满足的概率达到了 82%。根据以上结果，表明所提出的方法实现了高水平的通用性能，几乎可以满足各种场景下的切片需求。

图 8(a)和 8(b)显示了 RBUR 评估结果。所提出的方法的平均 RBUR 约为 0.77。对比方法的平均 RBUR 都在 0.7 之下，资源过度分配都超过了 30%。hard 方法的平均 RBUR 最低，因为 hard 切片将所有 RB 平均分配给各个切片。有的切片会存在过度分配。nvs 方法根据不同切片的数据速率请求进行分配。但是可能有的切片请求的速率比其他切片高太多，导致有的切片分配了过多资源从而 RBUR 很低。

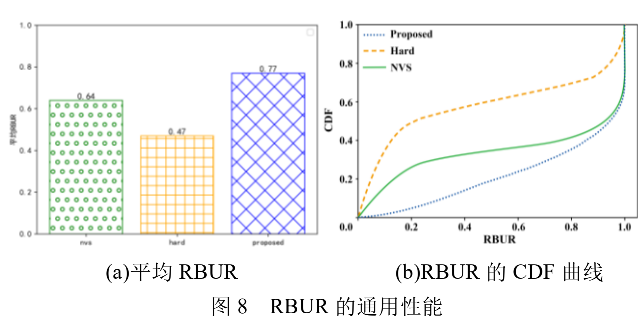
综上所述，对于随机生成的场景，所提出的方法能够有效地将 RB 分配给切片，几乎满足了各种切片的需求，实现了高水平的通用性能，同时资源利用率也很高。



(a)平均 NSDS　　(b)NSDS 的 CDF 曲线

图 7　NSDS 的通用性能

Fig. 7　General performance of NSDS



(a)平均 RBUR　　(b)RBUR 的 CDF 曲线

图 8　RBUR 的通用性能

Fig. 8　General performance of RBUR

3.3.3 扩展性评估

本文通过评估切片数量与性能之间的关系来评估所提出方法中切片数量的可扩展性。此评估场景使用第 3.2 节中使用的随机生成场景，切片数量根据每个场景的均匀分布从 1 到 8 中选择。评估时的切片数量不是根据场景创建时设置的切片数量来计算的，而是通过同时运行的切片数量来计算的。例如，如果切片数量设置为 7，但同时运行的切片数量为 3，则计为 3 个切片。评估指标为 NSDS 和 RBUR，测量结果为一秒内测量值的平均值。

图 9 显示了切片数量与 NSDS 的关系。对于所有测试方法，NSDS 随着切片数量的增加而降低。当切片数量增加时，需要更多的 RB。NSDS 减少是因为无法保证切片所需的 RB 数。当切片数量达到 4 个及以上时，所提出方法的 NSDS 比其他方法高出了 0.2 以上，实现了更好的效果。在切片数量少于 8 个时，NSDS 为 0.8 以上，切片中有 80%以上的用户完全满足了 QoS 需求。当切片数量为 8 时，NSDS 约为 0.78。

对比方法中，nvs 方法表现最差，因为一方面可能有的切片对于数据速率需求高，而有的切片对于数据速率需求低，导致对数据速率需求低的切片获得的 RB 数量过少，也就使得 NSDS 很低；另一方面，nvs 方法在基站之间均等分配资源，可能会使得在某个基站上分配过多资源，有的基站分配过少，导致用户接入的基站资源过少而降低了需求满意度。hard 方法将资源均等分配给各个切片，当切片数量增多时，每个切片获得资源变少，导致有的切片分配资源不足，需求满意度下降。

图 10 显示了切片数量和 RBUR 之间的关系。在所提出的方法中，当切片数量超过 4 个时，RBUR 随着切片数量的增加而降低，在切片数量为 8 时，RBUR 为 0.7。RB 分配过多是为了在分配 RB 时能够可靠地满足切片的需求。所提出的方法学习了将 NSDS 优先于 RBUR 的分配方式，原因和奖励的设计有关。在设计奖励时综合考虑了资源利用率和切片需求满意度。如果所有用户都不能满足需求，则 NSDS 为 0，而如果有用户使用了 RB，则 RBUR 就大于 0。奖励是 NSDS 乘以 RBUR，如果其中一个变为 0，则奖励也为 0。在学习时，接近 0 的 NSDS 会被优先考虑以改善切片的需求满意度，导致 RB 的过度分配。

在 nvs 方法和 hard 方法中，当切片数量为 1 时，hard 方法和 nvs 方法的 RBUR 不到 0.3，因为这两种方法会将所有的资源都分配给这个切片导致资源利用率过低，当切片数量增加时，会有更多的切片需要资源，也就使得切片的资源利用率得到了提高。但是因为这两种方法的局限性，RBUR 的表现都比所提出的方法要差，过度分配严重。

基于 NSDS 和 RBUR 的结果，即使切片数量发生变化，本文所提出的方法也能分配 RB 以满足切片对于吞吐量和时延的需求。此外，所提出的方法仅通过使用训练模型创建或终止执行 RB 分配的 actor 来应对切片数量的变化。综上所述，可以说所提出的方法在切片数量方面具有很高的可扩展性。

4　结束语

本文主要研究无线接入网(RAN)切片。在多基站环境下，为了能够有效地分配无线资源同时满足切片的需求，本文提出了一种使用 Ape-X 的 RB 分配方法，该方法不受切片数量的影响。仿真结果表明，所提出的方法能够根据切片的状态和需求分配资源，分配了必要数量的 RB 以满足切片的需求而且不受切片数量的变化的影响，同时对于随机生成的场景也实现了高水平的通用性能。

所提出的方法是基于 LTE 中的 TTI 设计的，在 5G 中，TTI 是可变的，未来会在这个方向上继续研究。

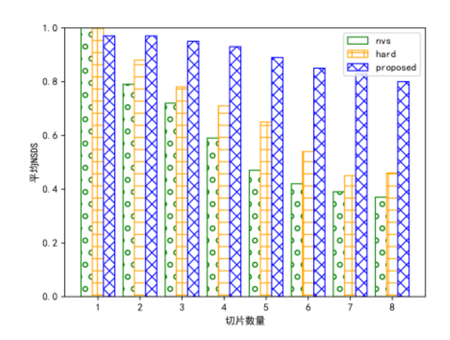


图 9　NSDS 扩展性评估

Fig. 9　NSDS scalability evaluation

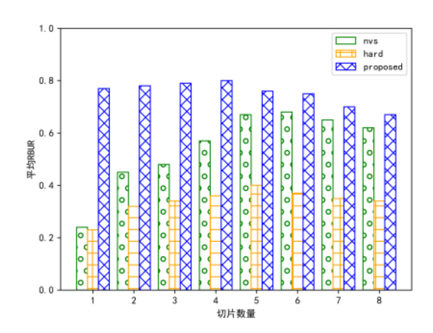


图 10　RBUR 扩展性评估

Fig. 10　RBUR scalability evaluation

参考文献：

[1] NGMN Alliance. 5G White Paper [EB/OL]. [2015-02-17]. https://www. ngmn. org/wp-content/uploads/NGMN_5G_White_Paper_V1_0. pdf.

[2] Elayoub S E, Fallgren M, Spapis P, *et al.* 5G service requirements and operational use cases: Analysis and METIS II vision [C]// 2016 European Conference on Networks and Communications (EuCNC) . Athens: IEEE, 2016: 158-162.

[3] Afolabi I, Taleb T, Samdanis K, *et al.* Network slicing and softwarization: A survey on principles, enabling technologies, and solutions [J]. IEEE Communications Surveys&Tutorials, 2018, 20 (3): 2429-2453.

[4] NGMN Alliance. Description of network slicing concept [EB/OL]. [2018-02-20]. http://www. ngmn. org/fileadmin/user_upload/160113_Network_Slicing_v1_0. pdf.

[5] Foukas X, Patounas G, Elmokash A, *et al.* Network Slicing in 5G: Survey and Challenges [J]. IEEE Communications Magazine, 2017, 55 (5): 94-100.

[6] Elayoubi S E, Jemaa S B, Altman Z, *et al.* 5G RAN Slicing for Verticals: Enablers and Challenges [J]. IEEE Communications Magazine, 2019, 57 (1): 28-34.

[7] Shrivastava R, Samdanis K, Bakry A. On Policy Based RAN Slicing for Emerging 5G TDD Networks [C]// 2018 IEEE Global Communications Conference (GLOBECOM) . Emirates, United Arab Emirates: IEEE, 2018: 1-6.

[8] Chang C Y, Nikaein N, Spyropoulos T. Radio Access Network Resource

Slicing for Flexible Service Execution [C]// IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) . Honolulu, HI, USA: IEEE, 2018: 668-673.

[9] Guo T, Suarez A. Enabling 5G RAN Slicing with EDF Slice Scheduling [J]. IEEE Transactions on Vehicular Technology, 2019, 68 (3): 2865-2877.

[10] 唐伦, 魏延南, 马润琳, 等. 虚拟化云无线接入网络下基于在线学习的网络切片虚拟资源分配算法 [J]. 电子与信息学报, 2019, 41 (7): 1533-1539. (Tang Lun, Wei Yannan, Ma Runlin, *et al.* Online Learning-based Virtual Resource Allocation for Network Slicing in Virtualized Cloud Radio Access Network [J]. Journal of Electronics & Information Technology, 2019, 41 (7): 1533-1539.)

[11] Abiko Y, Mochizuki D, Saito T D, *et al.* Proposal of Allocating Radio Resources to Multiple Slices in 5G Using Deep Reinforcement Learning [C]// 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE) . Osaka: IEEE, 2019: 131-132.

[12] 陈前斌, 施颖洁, 杨希希, 等. 基于在线双向拍卖的虚拟网络切片资源调度机制 [J]. 电子与信息学报, 2018, 40 (7): 1738-1744. (Chen Qianbin, Shi Yingjie, Yang Xixi, *et al.* Resource Scheduling Mechanism for Virtual Network Slice Based on Online Double Auction [J]. Journal of Electronics & Information Technology, 2018, 40 (7): 1738-1744.)

[13] Sun G L, Xiong K, Boateng G O, *et al.* Autonomous Resource Provisioning and Resource Customization for Mixed Traffics in Virtualized Radio Access Network [J]. IEEE Systems Journal, 2019, 13 (3): 2454-2465.

[14] Xiong K, Adolphe S Xiang R, Boateng G O, *et al.* Dynamic Resource Provisioning and Resource Customization for Mixed Traffics in Virtualized Radio Access Network [J]. IEEE Access, 2019, 7: 115449-115453.

[15] Kokku R, Mahindra R, Zhang H, *et al.* NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks [J]. IEEE/ACM Transactions on Networking, 2012, 20 (5): 1333-1346.

[16] Caballero P, Banchs A, Veciana G D, *et al.* Network slicing games: Enabling customization in multi-tenant networks [J]. IEEE/ACM Transactions on Networking, 2019, 27 (2): 662-675.

[17] Horgan D, Quan J, Budden D, *et al.* Distributed prioritized experience Replay [EB/OL]. [2018-03-02]. http://arxiv. org/abs/1803. 00933.

[18] Hasselt H V, Guez A, Silver D. Deep reinforcement learning with double Q-learning [EB/OL]. [2015-12-08]. http://arxiv. org/abs/1509. 06461

[19] Sutton R S, Barto A G. Reinforcement Learning: An Introduction [J]. IEEE Transactions on Neural Networks, 1998, 9 (5): 1054-1054.

[20] Wang Z, Schaul T, Hessel M, *et al.* Dueling network architectures for deep reinforcement learning [EB/OL]. [2016-04-5]. http://arxiv. org/abs/1511. 06581